



UNIT STATES DEPARTMENT OF COMMERCE
Patent and Trademark Office

Address: COMMISSIONER OF PATENTS AND TRADEMARKS
Washington, D.C. 20231

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.
-----------------	-------------	----------------------	---------------------

03/831,845 04/01/87 CALDEF

63 P2167/8UNIVPI

LM02/0105

Cla
EXAMINER

HICKMAN BEYER & WEAVER
PO BOX 61059
PALO ALTO CA 94306

BULLOCK JRL

ART UNIT

PAPER NUMBER

2755

Cla
DATE MAILED:

01/05/00

Please find below and/or attached an Office communication concerning this application or proceeding.

Commissioner of Patents and Trademarks

Office Action Summary

Application No. 08/831,845	Applicant(s) Calder, Bartley H. et al.
Examiner Lewis Bullock, Jr.	Group Art Unit 2755

Responsive to communication(s) filed on Oct 19, 1999.

This action is FINAL.

Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11; 453 O.G. 213.

A shortened statutory period for response to this action is set to expire 3 month(s), or thirty days, whichever is longer, from the mailing date of this communication. Failure to respond within the period for response will cause the application to become abandoned. (35 U.S.C. § 133). Extensions of time may be obtained under the provisions of 37 CFR 1.136(a).

Disposition of Claims

Claim(s) 1-23 is/are pending in the application.

Of the above, claim(s) _____ is/are withdrawn from consideration.

Claim(s) _____ is/are allowed.

Claim(s) 1-23 is/are rejected.

Claim(s) _____ is/are objected to.

Claims _____ are subject to restriction or election requirement.

Application Papers

See the attached Notice of Draftsperson's Patent Drawing Review, PTO-948.

The drawing(s) filed on _____ is/are objected to by the Examiner.

The proposed drawing correction, filed on _____ is approved disapproved.

The specification is objected to by the Examiner.

The oath or declaration is objected to by the Examiner.

Priority under 35 U.S.C. § 119

Acknowledgement is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d).

All Some* None of the CERTIFIED copies of the priority documents have been

received.

received in Application No. (Series Code/Serial Number) _____.

received in this national stage application from the International Bureau (PCT Rule 17.2(a)).

*Certified copies not received: _____.

Acknowledgement is made of a claim for domestic priority under 35 U.S.C. § 119(e).

Attachment(s)

Notice of References Cited, PTO-892

Information Disclosure Statement(s), PTO-1449, Paper No(s). _____

Interview Summary, PTO-413

Notice of Draftsperson's Patent Drawing Review, PTO-948

Notice of Informal Patent Application, PTO-152

--- SEE OFFICE ACTION ON THE FOLLOWING PAGES ---

Art Unit: 2755

DETAILED ACTION

Claim Rejections - 35 USC § 103

1. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

(a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negated by the manner in which the invention was made.

2. Claims 1-23 are rejected under 35 U.S.C. 103(a) as being unpatentable over “The Java Language Environment” by Sun in view of Skeen (US 5,187,787).

As to claim 1, Sun teaches a framework for associating data (object) with a command object (Java code to support object / compression algorithm), the command object being arranged to operate on the data, wherein the data is associated with an application (browser), comprising: a data retriever mechanism (browser) being arranged to obtain the data (object); and a mapping mechanism (browser) being arranged to obtain the command object (Java code to support object / new compression algorithm), wherein the command object is obtained by the mapping mechanism based substantially on the data (pg. 77-79, “HotJava can dynamically link the code from the host that has the image allowing it to display the new format. So, if someone invents a new compression algorithm, the inventor just has to make sure that a copy of its Java code is installed on the server that contains the images they want to publish...”; “Rather than having built-in protocol handlers, HotJava uses the protocol name to link in the appropriate handler as required, allowing new protocols to be incorporated dynamically.”). It would be obvious that the

Art Unit: 2755

downloaded object is mapped to the corresponding Java code or compression algorithm in order to allow some action to be performed on the object. However, Sun does not explicitly teach a data handler mechanism which is in communication with the application, the data retriever mechanism, or the mapping mechanism.

Skeen teaches a data handler mechanism (communication component / communication daemon) arranged to interface with the application (application 18 / application 16), the data retriever mechanism (directory services component retrieves service records / class manager), and the mapping mechanism (subject mapper / data-exchange component / forms manager module) (Col. 8, lines 47-60; Col. 16, line 62-Col. 18, line 7; Col. 18, line 20-Col. 20, line 35). It would be obvious that since the service discipline (command object) examines the service records (data) passed to it and determines the location of the service with which communications are to be established (Col. 19, lines 58-65), it therefore binds the service record with the service discipline. It would also be obvious that since the class manager retrieves the address of the class instance and the data exchange component converts data objects including their headers from one form to another, the stored data is therefore bound and mapped to be understood by the requesting application. Therefore, it would be obvious to modify the teachings of Sun with the teachings of Skeen in order to free applications from unnecessary information and facilitating modularization and flexibility (Col. 18, lines 20-47).

Art Unit: 2755

As to claim 9, Sun teaches a method for associating data (object) with a command object (Java code to support object / compression algorithm) in response to a request from an application (user / browser) by returning a command object associated with the data to the application (pg. 77-79, “HotJava can dynamically link the code from the host that has the image allowing it to display the new format. So, if someone invents a new compression algorithm, the inventor just has to make sure that a copy of its Java code is installed on the server that contains the images they want to publish...”; “Rather than having built-in protocol handlers, HotJava uses the protocol name to link in the appropriate handler as required, allowing new protocols to be incorporated dynamically.”). It would be obvious that the downloaded object is mapped to the corresponding Java code or compression algorithm in order to allow some action to be performed on the object. However, Sun does not teach the accessing steps, or the obtaining step and the binding steps performed in an interface.

Skeen teaches the steps of accessing the data (service records / address) through an interface (communication component / communication daemon) in response to the request from the application (subscribe call / Get_Field calls) (Col. 18, lines 48-51; Col. 16, line 62-Col. 17, line 58), the interface being independent from the application and in communication with the application, wherein the request from the application is processed by the interface (subject based addressing / data distribution decoupling); accessing a mapping mechanism (service discipline / subject mapper; data-exchange component; forms-manager module) which is in communication with the interface, the mapping mechanism being independent from the application (within

Art Unit: 2755

communication component), the mapping mechanism being arranged to locate a command object (service discipline; convert function) that is appropriate for the data (service record; data object), wherein the mapping mechanism is accessed by the interface (Col. 19, lines 47-65; Col. 8, line 47-Col. 9, line 12); and obtaining the command object that is appropriate for the data (subject mapper invokes the service discipline of that service; converting data objects; Get-data call after receiving field address), wherein the mapping mechanism obtains the command object and passes the obtained command object to the interface (service discipline in interface; converting operation performed in interface) (Col. 13, line 47-Col. 14, line 48). It would be obvious that the since the service discipline (command object) examines the service records (data) passed to it and determines the location of the service with which communications are to be established (Col. 19, lines 58-65), it therefore binds the service record with the service discipline. It would also be obvious that a service discipline (command object) can be returned to the application if there does not exist a subject mapper (Col. 20, lines 48-63). Refer to claim 1 for the motivation to combine.

As to claim 23, refer to claim 1 for rejection. However, claim 23 also details that the data handler mechanism interfaces with a plurality of applications and is independent from the plurality of applications so that data and a command object is obtained for a selected application. Skeen teaches the data handler mechanism (common communication daemon) interfaces with a plurality of applications (application 16 and application 18) and is independent from the plurality of applications so that data and a command object (data) is obtained for a selected application

Art Unit: 2755

(Col. 6, lines 53-55; Col. 21, lines 43-50; Col. 22, lines 6-20; Col. 27, lines 7-16). Refer to claim 1 for the motivation to combine.

As to claim 2, Skeen teaches data comprise of a stream of bytes (Col. 13, lines 25-31).

As to claim 3, Skeen teaches a data content handler mechanism (data-exchange component) being arranged to convert the data into a data object (data objects / forms) (Col. 8, lines 47-60).

As to claim 4, Sun teaches the data object (object) is created using the Java programming language and the command object is a Java command object (Java code to support object) (pg. 77-78).

As to claim 5, It would be obvious that the data would comprise of text or images.

As to claim 6, Skeen teaches the data handler (interface) is arranged to receive a request from the application (subscribe request; Get_Field request; Get_Data request) (Col. 18, lines 48-51; Col. 17, line 30-Col. 18, line 7). It would be obvious that the since the service discipline (command object) examines the service records (data) passed to it and determines the location of the service with which communications are to be established (Col. 19, lines 58-65), it therefore

Art Unit: 2755

binds the service record with the service discipline. It would also be obvious that a service discipline (command object) can be returned to the application if there does not exist a subject mapper (Col. 20, lines 48-63) otherwise the address can be passed through the call back routine. Sun also teaches the command object (Java code to support object) is returned to the application (browser / user) (pg. 78).

As to claim 7, Skeen teaches a data source mechanism (server / tables) arranged to obtain a stream of bytes (Col. 20, lines 9-12; Col. 17, lines 12-14; Col. 14, lines 38-48) and a data content handler mechanism (data-exchange component) arranged to convert the stream of bytes (data) into a data object (data object /forms), the data source mechanism being in communication with the data content handler mechanism (via the interface) (Col. 8, lines 47-60).

As to claim 8, It would be obvious that the mapping mechanism (service mapper) would have a look-up table in order to determine which service discipline to invoke (Col. 19, lines 58-65).

As to claim 10, Skeen teaches the steps of: passing a stream of bytes (information) to a data content handler mechanism (data-exchange component) arranged to create a data object (data object / forms) from the stream of bytes (data) (Col. 8, lines 47-60). It would be obvious that the data object is passed in the interface since the stream is converted in the interface.

Art Unit: 2755

As to claim 11, Sun teaches the data object (object) is created using the Java programming language, and the command object (Java code to support object) is a Java command object (pg. 77-78).

As to claim 12, Skeen teaches accessing a data retriever (directory services component retrieves service records / server / data-exchange component) which is arranged to obtain the data (service records / information / converted message) (Col. 18, line 20-Col. 20, line 35; Col. 13, lines 50-60). It would be obvious that the data could be a stream of bytes.

As to claim 13, Skeen teaches operating on the data using the command object (service discipline; conversion program) (set up communication / get information / convert data) (Col. 19, lines 58-65 ; Col. 20, lines 20-35; Col. 13, line 58-Col. 15, line 55).

As to claim 14, Sun teaches the command object (Java code to support object) that is appropriate for the data (object) is selected from a set of command objects associated with a command list (federation of pieces) (pg. 76, 78-79, “HotJava is the coordinator of a federation of pieces, each with individual responsibilities.”; “HotJava Browser is given a reference to an object. If the handler for that protocol is already loaded, it will be used. If not, the HotJava Browser will search first the local system and then the system that is the target of the URL.”). It would be obvious that Java code to support various objects would be kept in a list. It would also be

Art Unit: 2755

obvious that the teachings of Sun could be implemented in the interface of Skeen in order to hide communication details from the user.

As to claim 15, Skeen teaches the step of receiving a request for a command list from the application (subscriber request), the request for the command list being received by the interface (subject mapper within communication component), wherein the interface performs the steps of: obtaining a type (fields) associated with the data (service records); obtaining the command list through the mapping (all service disciplines that provided desired service); and returning the command list to the application (via call back routine) (Col. 18, line 41-Col. 20, line 63).

As to claim 21, Skeen teaches the command object (service discipline; conversion program) is obtained by the mapping mechanism (service mapper; forms manager) based substantially on the data (service record) without an external input from a user of the application (Col. 18, line 66- Col. 19, line 46; Col. 13, line 58-Col. 15, line 55). It would be obvious that since the service mapper obtains the service discipline from the service records obtained that the user had not input in this process.

As to claim 22, Skeen teaches the command object (service discipline; conversion program) is obtained by the mapping mechanism (service mapper; forms manager) based substantially on the data (service record; data) without directly involving the application (Col. 18,

Art Unit: 2755

line 66- Col. 19, line 46; Col. 13, line 58-Col. 15, line 55). It would be obvious that since the service mapper obtains the service discipline from the service records obtained that the application had not involvement in this process.

As to claims 16-20, reference is made to a computer program product which corresponds to the method of claims 9-11, 13, and 14 and is therefore met by the rejection of claims 9-11, 13, and 14 above.

Conclusion

3. Applicant's amendment necessitated the new ground(s) of rejection presented in this Office action. In particular, applicant amend claim 1 that the command object is obtained by the mapping mechanism based substantially on the data and claims 9 and 16 wherein the interface is independent of the application. Accordingly, **THIS ACTION IS MADE FINAL**. See MPEP § 706.07(a). Applicant is reminded of the extension of time policy as set forth in 37 CFR 1.136(a).

A shortened statutory period for reply to this final action is set to expire THREE MONTHS from the mailing date of this action. In the event a first reply is filed within TWO MONTHS of the mailing date of this final action and the advisory action is not mailed until after the end of the THREE-MONTH shortened statutory period, then the shortened statutory period will expire on the date the advisory action is mailed, and any extension fee pursuant to 37

Art Unit: 2755

CFR 1.136(a) will be calculated from the mailing date of the advisory action. In no event, however, will the statutory period for reply expire later than SIX MONTHS from the date of this final action.

4. Any inquiry concerning this communication or earlier communications from the examiner should be directed to Lewis A. Bullock, Jr. whose telephone number is (703) 305-0439.

Majid Banankhah
MAJID A. BANANKHAH
PRIMARY EXAMINER

lab

December 29, 1999